

まとめ

- データベースとは
 - 共有するデータを、便利で安全に管理できる場所
- データモデルとは
 - 記号系
 - ネットワーク, 階層, 関係, オブジェクト指向
- データベース管理システムとは
 - 効率的で信頼性高く, 便利に, かつ複数人で使用可能な, 大量の永続データの保存と利用を可能にする機能を提供するもの
- ファイルとの違い
 - DBIは「共有する」というための機能がいろいろ満載
- Whether you know it or not, you're using a database every day hour.
- 次回は2章

リレーショナルデータベース

- 主な応用
 - 企業内部のデータ管理
 - 在庫, 人事, 販売, 生産, 顧客情報など
- 「使いやすく, 問題が発生しにくい」が重要
 - データベースは大事なデータを管理する場所
- リレーショナルデータベース: しくみが単純
 - データが表形式で表せて簡単
 - 人間が出力を見たときに分かりやすい
 - それぞれの処理が明快
 - データを無矛盾に保ち, 障害にも対応

2

第2章 リレーショナルデータモデル —構造記述—

数学の集合論に基づいたリレーショナルデータベースの構造記述

3

今回の内容

- リレーショナルデータモデル
 - 定義に使われている数学的用語
 - リレーション, テーブル
 - スキーマ
 - インスタンス, タプル
 - 第1正規形
 - 空値

4

リレーショナルデータモデルの提案

- E. F. Codd, "A Relational Model of Data for Large Shared Data Banks," Communications of the ACM, Vol.13, No.6, pp.377-387, 1970.
- アプリケーションプログラムとデータ(に關係すること)を切り離す＝データ独立性

5

A Relational Model of Data for Large Shared Data Banks

E. F. Codd
IBM Research Laboratory, San Jose, California

Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed. Changes in data representation will often be needed as a result of changes in query, update, and report traffic and natural growth in the types of stored information.

Existing noninferential, formatted data systems provide users with tree-structured files or slightly more general network models of the data. In Section 1, inadequacies of these models are discussed. A model based on n -ary relations, a normal form for data base relations, and the concept of a universal data sublanguage are introduced. In Section 2, certain operations on relations (other than logical inference) are discussed and applied to the problems of redundancy and consistency in the user's model.

6

リレーショナルデータモデル

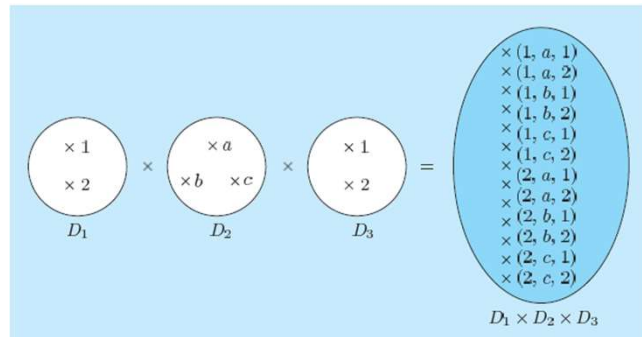
- 従来のデータモデルと比べて単純, データ独立性が高い
 - 複数の属性の組合せによって表を定義
- 表同士の演算によりさまざまな処理を実現
- (可変長のデータや, 構造が複雑なデータには不向き)
- 単純＝数学的

7

リレーション(数学的用語で定義)

- ドメイン: データの定義域
 - 人名, 年令, 給与
- 直積
- タプル
- リレーション: ドメイン D_1, D_2, \dots, D_n 上のリレーションとは, これらドメインの直積 $D_1 \times D_2 \times \dots \times D_n$ の任意の有限部分集合
- 濃度(基数): タプルの総数
- 次数: リレーションが定義されているドメインの数

8



次数3=3項(ternary)リレーション

次数がnのときは, n項(n-ary)リレーション

9

リレーションのテーブル表現

1	a	1
1	a	2
2	b	1
2	c	2

行(row):タプル
列(column)

行の並び順は, 何の情報も担っていない
列の順番には意味がある

10

属性名とリレーション名

$D_1 = \{x \mid x \text{は人名}\}, D_2 = \text{INTEGER}$

リレーションR

$R \subseteq D_1 \times D_2$ (\subseteq は部分集合の意味)

$R = \{(\text{太郎}, 25), (\text{一郎}, 30), (\text{花子}, 26), (\text{桃子}, 22)\}$

友人	
名前	年齢
太郎	25
一郎	30
花子	26
桃子	22

テーブル表現

友人	
名前	年齢
太郎	25
一郎	30
花子	26
桃子	22

名前付きテーブル表現

11

属性名とリレーション名

リレーション $R (\subseteq D_1 \times D_2 \times \dots \times D_n)$ の属性名を A_1, A_2, \dots, A_n とし, ドメイン関数 dom を次のように定義

$\text{dom}: A_i \rightarrow D_i (i=1, 2, \dots, n)$

すると, $R \subseteq \text{dom}(A_1) \times \text{dom}(A_2) \times \dots \times \text{dom}(A_n)$

例: $\text{友人} \subseteq \text{dom}(\text{名前}) \times \text{dom}(\text{年齢})$

友人	
名前	年齢
太郎	25
一郎	30
花子	26
桃子	22

リレーションスキーマ
インスタンス

12

リレーション(数学的用語で定義)

- ドメイン: データの定義域
 - 人名, 年齢, 給与
- 直積
- タプル
- リレーション: ドメイン D_1, D_2, \dots, D_n 上のリレーションとは, これらドメインの直積 $D_1 \times D_2 \times \dots \times D_n$ の任意の有限部分集合
- 濃度(基数): タプルの総数
- 次数: リレーションが定義されているドメインの数

13

スキーマとインスタンス

- スキーマ(Schema)
 - リレーション名, リレーションの各属性名, ドメイン
 - キー制約, 参照整合性制約, 属性制約などの制約
 - リレーションの時間に不変な性質
- インスタンス
 - 属性値の組によって表されるデータ(タプル)の集合
 - 一般に時間の移り変わりと共に変化

従業員

従業員番号	部門番号	氏名	年齢
001	1	熊本太郎	27
002	2	熊本一郎	17
003	3	熊本花子	30

スキーマ
※ 実際には各種制約を含む

インスタンス

14

スキーマとインスタンス

- スキーマ
 - データベースに格納されるデータのデータ構造, データの型, データ同士の関連, 各種制約を記述したもの
 - 最初に一度定義したら(基本的に)変更されない
- インスタンス
 - スキーマに基づいて格納された実際のデータ
 - データベースの利用に応じて追加・削除・変更される

15

リレーションの例

科目

科目番号	科目名
01	データベース
03	コンピュータグラフィックス
...	...

学生

学籍番号	氏名
0123001	熊本太郎
0123002	熊本一郎
0123003	熊本花子
...	...

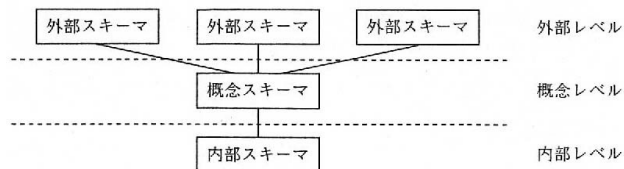
履修

科目番号	学籍番号	成績
01	0123001	60
03	0123002	80
01	0123003	70
...	...	

16

抽象化の3レベル

- DBMSの管理するスキーマの3階層
 - 外部スキーマ, 概念スキーマ, 内部スキーマ
 - ANSI/SPARCモデル[1975](下図)



17

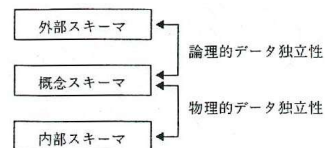
抽象化の3レベル

- DBMSの管理するスキーマの3階層
 - 内部スキーマ
 - 物理レベルのデータ構造
 - 概念スキーマ
 - データベースの定義言語で定義されるスキーマ
 - 外部スキーマ
 - 外部のアプリケーションから操作する時に, それぞれのアプリケーションインターフェイスに見せるスキーマ

18

データの独立性

- データベースではデータはアプリケーションとは独立して組織化される
 - 論理的データ独立性
 - 外部スキーマ・アプリケーションには影響を与えない範囲で**概念スキーマ**を変更可能
 - 物理的データ独立性
 - 概念データスキーマに影響を与えない範囲で,
内部スキーマ(=物理データ構造)を変更可能



19

データベースの閉世界仮説

- closed-world assumption
- 現時点で真であると判明していないことは偽とする
- 現時点でデータベースに記録されていない事象は実世界で生起していない
- データベースには, データが過不足なく存在

20

第1正規形

- リレーション(スキーマ)を定義するすべてのドメインがシンプルのとき、そのリレーションは第1正規形(the first normal form, 1NF)であるという。
 - どの属性の値も集合であることはない
 - 複数の属性にまたがる値はない
- 入れ子型リレーション(nested relation)や、べき集合の属性があってはならない
- ドメインがシンプルでないリレーションは非第1正規形(non-first normal form, (NF)²)

21

非第1正規形リレーションの例

社員

社員番号	社員名	...
0650	(鈴木, 一郎)	
1508	(浜崎, アユ)	⋮

入れ子リレーション
nested relation

社員

社員番号	社員名	趣味	...
0650	鈴木一郎	{野球, 盆栽, コイン収集}	⋮
1508	浜崎アユ	{作詞, ショッピング}	

べき集合
2趣味

22

非第1正規形リレーションの例

社員

社員番号	社員名	...
0650	(鈴木, 一郎)	
1508	(浜崎, アユ)	⋮

社員

社員番号	社員名		...
	姓	名	
0650	鈴木	一郎	
1508	浜崎	アユ	⋮

23

非第1正規形リレーションの例

社員

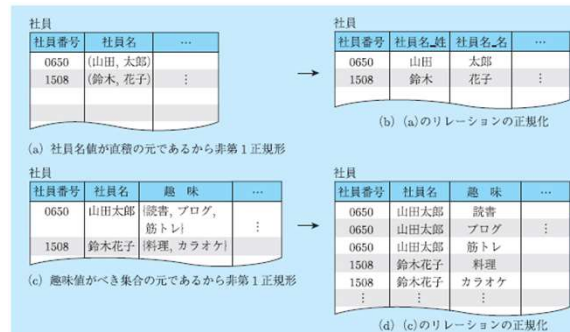
社員番号	社員名	趣味	...
0650	鈴木一郎	{野球, 盆栽, コイン収集}	⋮
1508	浜崎アユ	{作詞, ショッピング}	

社員

社員番号	社員名	趣味	...
		野球	
		盆栽	
		コイン収集	
1508	浜崎アユ	作詞	
		ショッピング	

24

非第1正規形リレーシヨンの正規化



25

Akifumi Makinouchi. 1977. A consideration on normal form of not-necessarily-normalized relation in the relational data model. In Proceedings of the third international conference on Very large data bases - Volume 3 (VLDB '77). VLDB Endowment, 447-453.

A CONSIDERATION ON NORMAL FORM OF NOT-NECESSARILY-NORMALIZED RELATION IN THE RELATIONAL DATA MODEL

Akifumi Makinouchi

Department of Computer Science, Fujitsu Laboratories, Ltd.
1015 Kamodanaka, Nakahara-ku, Kawasaki, Japan

In this paper definitions of unnormalized relation, functional dependency on it and Normal Form are presented. The Normal Form plays a key role in our relational data model in which unnormalized relations are admitted as does the Third Normal Form of Codd in the data model of normalized relation. Properties pertaining to Normal Form are discussed emphasizing comparison with 3NF along with analysis of some typical examples.

Similarity and dissimilarity between new functional dependency and Fagin's multivalued dependency are also discussed and presented in the form of proposition.

1. INTRODUCTION

Codd proposed a relational model of data in his earlier paper [1] and developed his idea on normalization of relational data model in his succeeding papers [2 and 4]. He defined three types of normal forms of relation, namely the First, Second and Third Normal Forms.

He introduced these Normal Forms, particularly the Third Normal Form, so that certain anomalies of data which are caused when storage operations such as insertion, deletion and update of data are applied to data base [3] can be avoided.

Recently, some researchers [5 and 6] found certain cases of data where the Third Normal Form does not work as nicely as Codd might have expected. To cope with these cases, R. Fagin added a new normal form to the series of normal forms by defining multivalued dependency in place of functional dependency, which is the basic idea in the definition of the normal forms of relation.

Both the 3NF (the Third Normal Form) by Codd and the 4NF (the Fourth Normal Form) by Fagin are defined on the relations of 1NF (the First Normal Form); that is, 3NF and 4NF imply 1NF, although the relation in

2. AN EXTENSION OF FUNCTIONAL DEPENDENCY

2.1 Definition of relation

Mathematically, a relation is defined as follows: Given sets D_1, D_2, \dots, D_n (not necessarily distinct), R is a relation on these n sets if it is a set of n -tuples (d_1, d_2, \dots, d_n) such that d_i belongs to D_i , where $i=1, 2, \dots, n$. D_1, D_2, \dots, D_n are domains of R .

In the realm of relational data model, a relation as above looks like a table (array) each of whose columns has different names. In the sequel, we use tables and relations interchangeably, but domains and columns differently. Each column name of a relation represents a role name in the relation and its domain is a set of values which may possibly be inserted into the column. Each row of the relation consists of an n -tuple of values.

When the domain of every column of a relation is a set of such values that they can be represented by an integer, a character string, etc., we call it a normalized relation. Such a value is said to be atomic and a column whose domain is a set of atomic values is called an atomic column.

26

空値 (4.4)

- 空値
 - 属性値が存在しないことを示す特殊な属性値
 - 「NULL」と表す
- ゼロや空白と空値の区別に注意
 - ゼロや空白の場合は、その属性値がゼロあるいは空白と分かっているもの
 - NULLの場合は、値が存在しない、値がまだ分からない、などを意味する

履修

科目番号	学籍番号	成績
001	00001	90
001	00002	0
001	00003	NULL

※ NULLは、値がないことを表す (例えば成績が未評価など)

27

まとめ

- リレーシヨナルデータモデル
 - 定義に使われている数学的用語
 - リレーシヨン, テーブル
 - スキーマ
 - インスタンス, タプル
 - 第1正規形
 - 空値
- 次回は第3章

28